THE

Edsger W. Dijkstra

Introduction

In response to a call explicitly asking for papers "on timely research and development efforts," I present a progress report on the multiprogramming effort at the Department of Mathematics at the Technological University in Eindhoven.

Presented at an ACM Symposium on Operating System Principles, Gatlinburg, Tennessee, October 1–4, 1967.

Goal of this system

Goal of this system

- “The primary goal of the system is to process smoothly **a continuous flow of** user programs as a service to the University”

The hardware

(1) core memory cycle time 2.5μsec, 27 bits; at present 32K;
(2) drum of 512K words, 1024 words per track, rev. time 40msec;
(3) an indirect addressing mechanism very well suited for stack implementation;
(4) a sound system for commanding peripherals and controlling of interrupts;
(5) a potentially great number of low capacity channels; ten of them are used (3 paper tape readers at 1000char/sec; 3 paper tape punches at 150char/sec; 2 teleprinters; a plotter; a line printer);

Outline

- Major design decisions
  - CPU management
    - Multi-programming
    - Processes
    - Synchronization
  - Memory management
    - Virtual memory
- Overall OS architecture
  - Monolithic kernel design

Multi-programming

- What does this mean?
  - Multiple programs running at *the same time*

Multi-programming

- What is its benefit over uni-programming?
  - Throughput!!!

Multi-programming

- What is its benefit?

versity. A multiprogramming system has been chosen with the following objectives in mind: (1) a reduction of turn-around time for programs of short duration, (2) economic use of peripheral devices, (3) automatic control of backing store to be combined with economic use of the central processor, and (4) the economic feasibility to use the machine for those applications for which only the flexibility of a general purpose computer is needed, but (as a rule) not the capacity nor the processing power.

Multi-programming

- What is its benefit?
  - Reduce turn-around time
  - Better utility of peripheral devices
  - Better utility of storage
  - Better accommodate for low-demand tasks

Multi-programming

- What is its benefit?
  - Reduce turn-around time
  - Better utility of peripheral devices
  - Better utility of storage
  - Better accommodate for low-demand tasks
- Does it have problems?
  - Switching among applications (processes) takes time

Memory management

- What is the difference between physical address and virtual address?

- char* x = malloc (10)

  *x =‘a’;

- printf(“Address of variable G is %d\n”, &G)

Memory management

- OLD days

Storage Allocation. In the classical von Neumann machine, information is identified by the address of the memory location containing the information. When we started to think about the automatic control of secondary storage we were familiar with a system (viz. GIER ALGOL) in which all information was identified by its drum address (as in the classical von Neumann machine) and in which the function of the core memory was nothing more than to make the information "page-wise" accessible.

Memory management

- THE

We have followed another approach and, as it turned out, to great advantage. In our terminology we made a strict distinction between memory units (we called them "pages" and had "core pages" and "drum pages") and corresponding information units (for lack of a better word we called them "segments"), a segment just fitting in a page. For segments we created a completely independent identification mechanism in which the number of possible segment identifiers is much larger than the total number of pages in primary and secondary store. The segment identifier gives fast access to a so-called "segment variable" in core whose value denotes whether the segment is still empty or not, and if not empty, in which page (or pages) it can be found.

What is the benefit of virtual memory?

What is the benefit of virtual memory?

- No need to write back to the same disk location
- A program has no need to occupy consecutive drum pages
- Making programming much easier

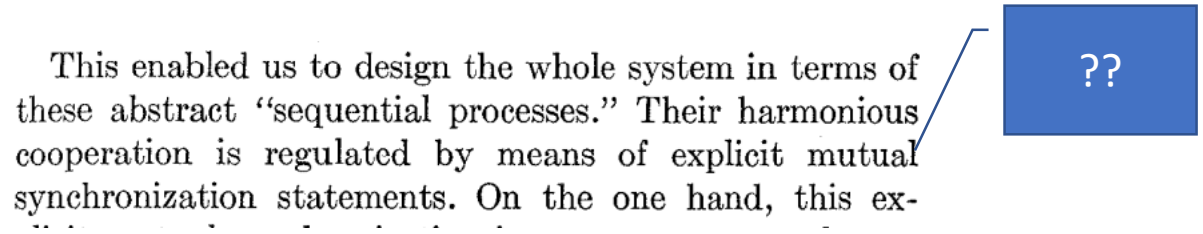
How is this achieved in THE?

The system does not cater for user programs written in machine language.

Processes

- What is the modern definition of “process”?
  - Running program
- What is the difference between processes in THE and Nucleus?
  - THE: no dynamic process creation; batching OS
  - Nucleus: start
- How does THE perform CPU scheduling?

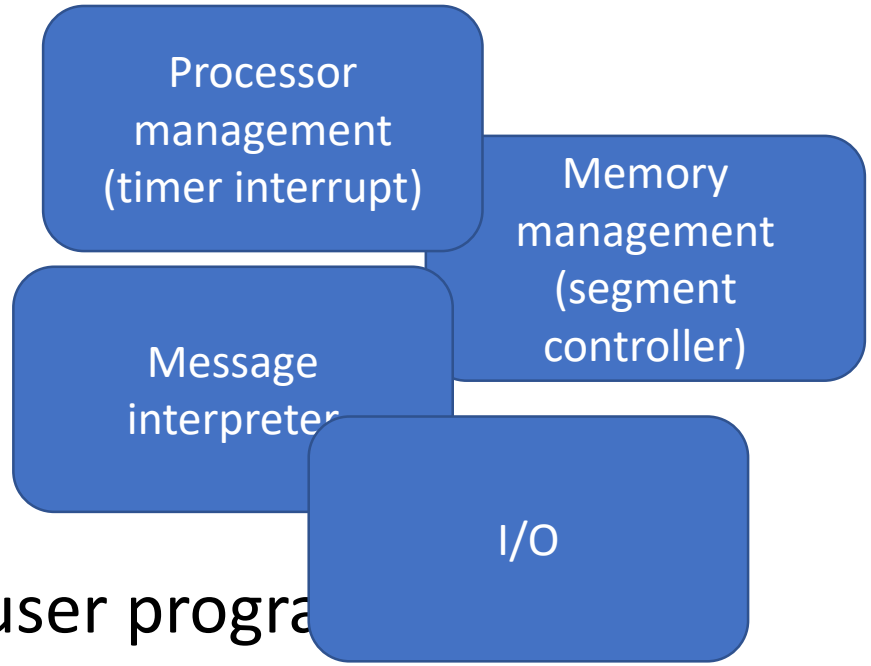
Processes & Synchronization



Is there a file system?

OS organization: monolithic kernel!!

- Level 0:
- Level 1:
- Level 2:
- Level 3:
- Level 4: user progra



Why levels?

- What exactly are these levels
- Why levels?

Nucleus

Goal of this system?

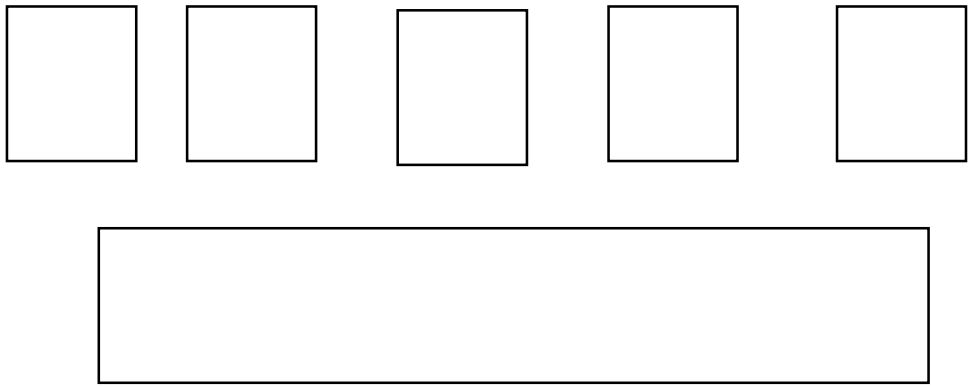
- Extensible

Goal of this system?

For the designer of advanced information systems, a vital requirement of any operating system is that it allow him to change the mode of operation it controls; otherwise his freedom of design can be seriously limited. Unfortunately, this is precisely what present operating systems do not allow.

This unfortunate situation indicates that the main problem in the design of a multiprogramming system is not to define functions that satisfy specific operating needs, but rather to supply a system nucleus that can be extended with new operating systems in an orderly manner. This is the primary objective of the RC 4000 system.

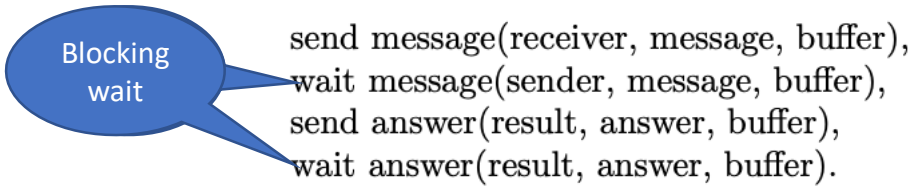
OS architecture: micro kernel



What is the minimum functionality a kernel should offer?

- Process creation (very different from THE!)
  - Start, Stop, Remove

- Process communication (very different from THE!)
  - A lot of security protection here!



Implementing different policies using nucleus mechanisms

- How to implement different scheduling policy?

Implementing different policies using nucleus mechanisms

- How to implement different memory management policy?

THE vs. Nucleus

	THE	Nucleus
Multi-programming?	Yes	Yes
dynamic processes?	No	yes
Memory management	VM	X
Synchronization	Semaphore	Message passing
OS architecture	Monolithic	Micro-kernel
Reliability		Better
Extensibility		Better
Performance	Better	